Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 13 (2012) 43 – 52

Procedia Computer Science

Proceedings of the International Neural Network Society Winter Conference (INNS-WC 2012)

Controlling Generative Processes of Generative Art

Somnuk Phon-Amnuaisuk[a,*], Jirapat Panjapornpon[b]

[a] *Faculty of Creative Industries, Universiti Tunku Abdul Rahman, Malaysia*
[b] *ThaiCED, Thailand*

Abstract

Computers have now become a crucial component in the process of digital media contents creation. Computer programs have been written to generate artistic artifacts, for examples, poetry, painting and music. These artifacts could be classified along the spectrum of algorithmic complexity of the programs, where order is at one end and disorder is at the other end. The artifacts classified toward the order end of the spectrum possess a clear structure (e.g., symmetry and tiling) while the artifacts classified toward the disorder end do not have any structure at both local and global levels (e.g., randomisation). Highly ordered or disordered generative art artifacts are generated from efficient algorithms that are simpler than those used to generate artifacts classified as lying between the order and disorder extremes. Control is embedded in the programs and it expresses the intention and strategy of the creative process. In this paper, we investigate the issue of control in generative art. We argue that the control expressed in the programs is a crucial component in a generative process. Hence, the ability to exert control is important in guiding the creative processes to intentionally generate complex and interesting artifacts. We describe the nature of the control observed in the generative process of computer generative art techniques. We then present examples of computer generated painting and discuss the control employed in the generative processes.

Keywords: generative art, controlling generative process, creative process

1. Introduction

Let us first define the notion of generative arts referred to in this paper. Among many definitions of generative art, we found the following definitions useful:

> Generative art refers to art that has been generated, composed, or constructed in an algorithmic manner through the use of systems defined by computer software algorithms, or similar mathematical or mechanical or randomised autonomous processes. (quote from http://www.all-art.org/artists-a-art_generative.html, last retrieved on 15 May 2012)

> Generative art refers to any art practice where the artist uses a system, such as a set of natural language rules, a computer program. a machine, or other procedural invention, which is set into motion with some degree of autonomy contributing to or resulting in a completed work of art [1].

*Corresponding author
Email address: span.amnuaisuk@gmail.com, somnuk@utar.edu.my (Somnuk Phon-Amnuaisuk)

1877-0509 © 2012 Published by Elsevier B.V. Selection and/or peer-review under responsibility of Program Committee of INNS-WC 2012
Open access under CC BY-NC-ND license. doi:10.1016/j.procs.2012.09.112

There is a consensus view that *process* is a necessary component for *generative art*. The term process used here carries the connotation of the stages of an operation and is not limited to the computing process in computers. Some control efforts might be deliberately delegated to outside autonomous processes. This meaning of process subsumes the notion of algorithm as algorithm is a well-defined process. Hence, it should be quite clear that control naturally resides in any algorithmic process. If the generative art is created using a computer program, the *control structure of a programming language* and *the ability to modify parameters relevant to the process which affects behaviours of the process* are examples of the control. These two control categories allow us to incorporate ideas and decisions into the creative process. In our stance, creativity must exist with intention and control is important for intentionally creating creative artifacts.

Let us formally define the notion of control used in this paper, let $\mathcal{A}$ be an algorithm constructed using a programming language $\mathcal{L}$. A semantic of a creative process could be read from the control structure chunk i, C_i in the program. These control chunks render a creative artifact according to the process set by us. Each control C_i is constructed using the primitives provided in the language $\mathcal{L}$ e.g.,

$$< control > ::= < expression > \mid < control >< operator >< expression >,$$

where $< control >, < operator >$ and $< expression >$ are nonterminal symbols which, at the end of the abstraction, are constructed from terminal symbols provided in the language $\mathcal{L}$. Since there are many ways to construct different control structures for the same behaviour e.g., 1+2+...+10 = 55 and (10+1)*(10/2) also gives 55, the complexity of algorithms is compared using the most efficient versions. The complexity of an efficient algorithm is determined from its space and time resource requirements. An algorithm A is more complex than an algorithm B if they are both compared using their most efficient versions and the algorithm A requires more resources than the algorithm B.

According to [2], creativity is defined as the production of an idea, action, or object that is new and valued. It is romantic to think that creativity is handed down to us on a good day from the divines (especially when the impending deadlines are near). However, evidence seems to suggest that creativity involves a lot of hard work. From the literature, most researchers agree that creativity cannot be created from void [2]. There must be some starting materials to begin with. George [3] argued that creativity is the ability to solve problems by generating new ideas which do not follow from the problem by formal deductive steps. Boden [4] argued that creativity is the mapping, exploration and transformation of conceptual spaces. Novel ideas and solutions emerge as a result of redescription, reconstruction, and transformation of existing materials. Unfortunately, the very nature of these activities (i.e., redescription, reconstruction, and transformation) is not well-understood yet. Wallas suggested that creativity is a process and differentiated four stages in the creative process: (i) preparation, (ii) incubation, (iii) insight and (iv) elaboration [5]. Perhaps, with the influneces from Artificial Intelligence research, many researchers attempt to view creative process from a computational perspective [6, 7, 8, 9]. Boden [4] agreed that computational ideas can help us understand how human creativity is possible.

In this paper, we take creativity as a computational process in which control is an important component. This paper aims to discuss the issue of control from the perspective of generative arts. In particular, we discuss the issues of controls employed in the generation of generative visual arts (e.g., abstract patterns, painting, etc.). Depending on the knowledge representation primitives, control could be compactly represented as adjustable parameters in mathematical equations, production rules or embedded control in program structures. This paper explores this control issue by providing a critical discussion and presents generative examples from various controls. The rest of the paper is organised as follows: Section 2 discusses the control component in computational generative art; Section 3 discusses the nature of the control employed in four examples of generative art; and the conclusion is provided in Section 4.

2. Control in Generative Art

Generative visual art and sound art were explored long before the birth of electronic computing machines. The creation and design of motives found in a garment are examples of control expressed in terms of *rules*. The Mozart's Musical Dice Game (Musikalisches Würfelspiel) was an example of generative music. Motives of geometric patterns, tiling patterns found in garment; ornamentation motives found in old buildings could be

considered as generative visual art. The act of rolling a dice is the control that the composer delegates to the system in the Mozart's Musical Dice Game. This is an example of a *stochastic process*. It is demonstrated in [10] that the harmonisation of Bach chorale could be carried out by computers using pre-composed control structures (built from a specialised control language).

Flake discussed the notion of effective complexity and pointed out that strictly regular and strictly irregular artifacts are *simple* [11]. Following the same line of thought, Galanter described generative art systems in the context of effective complexity [1]. He categorised generative art systems such as those based on symmetry and tiling as highly ordered (low complexity); generative art systems based on randomisation as highly disordered (also low complexity); and generative art systems based on evolutionary computation and artificial life approaches as lying between the order and disorder spectrum and having the most effective complexity. In this paper, we describe the control along the spectrum of *order* and *disorder* as well. Since control is part of a program, it follows that complex programs have complex control and simple programs have simple control.

In our stance, creativity must be with intention and control is the important part of an intentional creative process. Examining the nature of control in generative art could reveal different viewpoints and ideas on how to incorporate control into the creative process. Since creativity is a process that changes the structure of knowledge to produce novel artifacts [2] (i.e., via redescription, reconstruction, and transformation), a model for creating generative art could be created as a knowledge-based system in which an artifact could be produced/reconstructed by means of search (and search can also be seen as a problem-solving process or solution construction process). The major challenge in modelling computational creativity in this perspective is that we do not have a complete aesthetic theory to come up with a perfect objective function to guide search. Research in aesthetic evaluation has been quite active in the recent decade (see [12, 13, 14, 15]). However, most generative visual art still could not fully benefit from the findings from aesthetic evaluation. The majority body of generative art is produced using a stochastic process where randomness is the control strategy (see Figure 1); an interactive evolutionary process where users' feedback is used to control the creation of artifacts [16]; an ecological process where the behaviours of the system emerge from the interactions among agents in an ecological system [17, 18]; a world model (e.g., AARON) where the system possesses knowledge of shape, color, and dimension of objects to be drawn; and a mathematical model which could be an abstract representation of concepts.

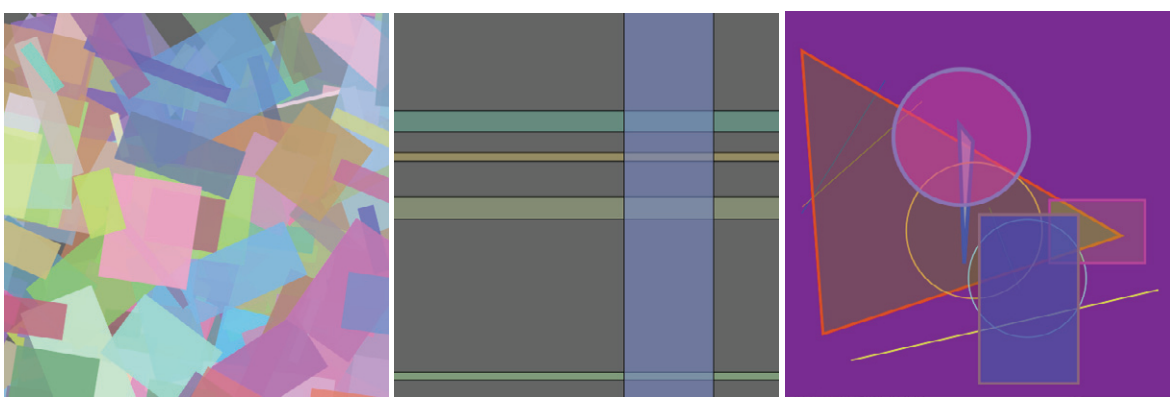


Fig. 1. Three images generated with stochastic process. Left-pane: rectangles of various sizes, colours and rotations are randomly composited on a canvas; Middle-pane: a random composition of strips of various random sizes and colours; Right-pane: a random composition of lines and other geometrical objects.

2.1. Generative Models Based on Stochastic Process

Stochastic process has been successfully employed to generate many abstract visual artifacts. Abstraction is the process that reduces unnecessary details in order to show the crust of the idea. Abstract art does not intend to present any narration although one may argue that abstract visual artifacts allow us to visualise the concept of randomness. Figure 1 shows three outputs from randomly drawn shapes with different parameters (e.g., width, height, radius, opacity, etc). These examples are created by programs that have been intentionally written to

imitate works from the *suprematism* and *neo-plasticism* movements [19]. In our opinion, the images do reflect the flavour of suprematism and neo-plasticism to a certain degree.

2.2. Generative Models Based on Production Rules and Grammar

Generating artifacts using stochastic process could produce endless interesting abstract images. However, it is often not possible to manipulate the dependency and relationship among components (in the image) whether in a local or a global structure. In the examples given in Figure 1, it is not possible to control the position, size or colour of the triangles. To gain more control, control information must be expressed in the program. *Production Rules* and *Grammar* are common tactics employed to encode control and knowledge about world objects in the generative model.

Production rules are often expressed in terms of *IF–THEN* or *IF–THEN–ELSE* statements. It is plausible to encode knowledge about world objects such as the physical properties of a tree, a table, a chair, a boy, a male adult, a dog, etc. in the rules. This knowledge base can be used in the generative process. The AARON system can produce interesting line-drawing with a specific intention, for example, people in various poses (see Figure 2 left-pane).

Fig. 2. Left-pane: a painting from the AARON system (an example from www.kurzwei/cyberart.com/aaron/static.html); and Right-pane: a graphic generated from the software tool *Virtual Laboratory* (algorithmicbotany.org)

Grammar provides a compact way to pack information. Linguists employ grammar to compactly express syntactical structure of a language. In 1968, Lindenmayer introduced a new style of string rewriting mechanisms. The rewriting is carried out simultaneously in parallel [20]. This approach is known as the L-system and it has been successfully employed to model various plants. Figure 2 (right-pane) shows an example of a plant produced from the L-system [1].

2.3. Generative Models Based on Evolutionary Computing and Ecosystemic Approach

Evolutionary computing has been widely explored as a generative technique. An informative review of research in this area could be found in [21]. In this paradigm, plausible solutions or partial solutions are coded as a chromosome population. Each chromosome represents a position in the search space. Evolution mechanisms allow these chromosomes to breed and produce offsprings. At the end of each generation, fitter chromosomes survive and weaker chromosomes are removed from the population. This kind of reproduction pressure heuristically guides search to a better solution. Control of search direction is therefore from the reproduction mechanisms (e.g., mating criteria, fitness evaluation criteria).

Genetic Algorithm (GA) and Genetic Programming (GP) are common techniques used to create abstract images (e.g., breeding mathematical formula). It is common to code chromosome population using mathematical

[1] This is an example image given in the software tools for performing simulated experiments: the *Virtual Laboratory*, obtained from algorithmicbotany.org

formula. Evolution is often guided by interactive users' feedback since this is often the most effective way as it is impractical or even impossible to come up with a suitable aesthetic evaluation function. Another influential direction in the evolutionary theme is generative art systems based on Evolutionary Artificial Life. Many informative discussions and interesting 2D and 3D images can be found in [22, 23, 24].

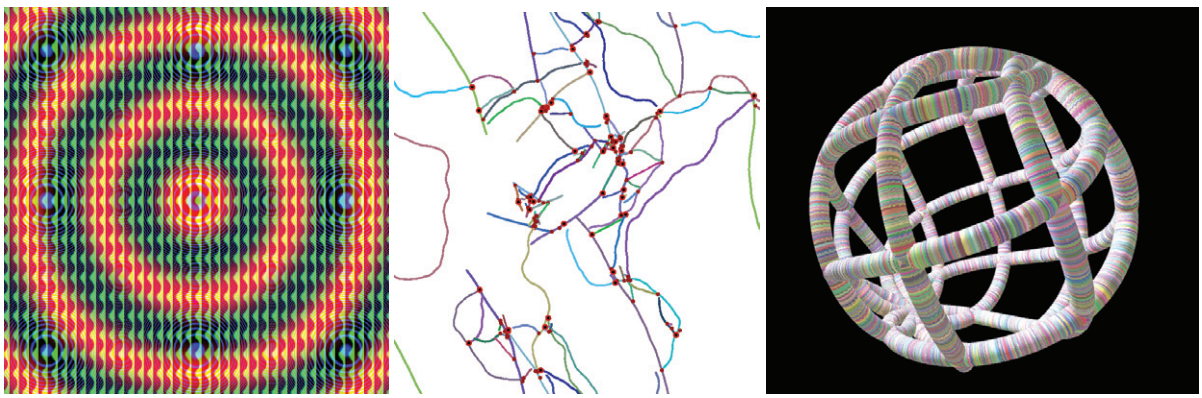


Fig. 3. Three images generated with different styles of control. Left-pane: an abstract pattern generated from mathematical equations; Middle-pane: traces of the movements of particles are plotted. A particle interacts with the environment, breeds (creating a new particle when the reproduction conditions are right) and dies when colliding with other particles; and Right-pane: a 3D model generated by tracing the particles' movements in a 3D space.

The ecosystemic approach [17, 25, 18] has gained more interest from the generative art community in recent years since it provides an alternative to the interactive evolutionary approach. The ecosystemic approach does not require fitness functions and the system behaviours emerge from the properties of the system itself. An ecosystem can be seen from the multiagent perspective where agents interact among themselves and the environment. These interactions give rise to overall complex behaviours of the system. In generative art, these behaviours can be translated into drawings e.g., the trajectory of agents in the environment. Figure 3 presents three examples created using three different control styles based on concepts discussed in this section.

3. Case Studies

In the previous section, we gave an overview of various controls and presented artifacts generated using various control techniques. In this section, we detail four generated artifacts that exploit the concepts of ecosystem, stochastic process and rules. These concepts are weaved together as a program. This allows us to create a control structure that exploits various concepts to generate artistic artifacts.

3.1. A Generative Example from an Ecosystemic Approach

Figure 4 is generated using the ecosystemic approach. The ecosystemic system is a dynamic system capable of producing complex patterns. In this example, the painting is generated from the behaviours of the agents in which their interactions with the environment are expressed in deterministic rules. Twelve agents are created with random stroke colours and sizes. Once created, they are randomly placed on the canvas. Each agent moves in circle and reproduces when the conditions are right (dependent on its energy and population density of the agents in its neighborhood). The reproduction creates a new agent branching out from the parent agent. The trajectories of the agents are plotted as they traverse the environment. The agents need energy to move and will die after depleting their energy or after colliding with other agents. The pseudo code described in Figure 4 gives more detailed information of the control employed in generating this image.

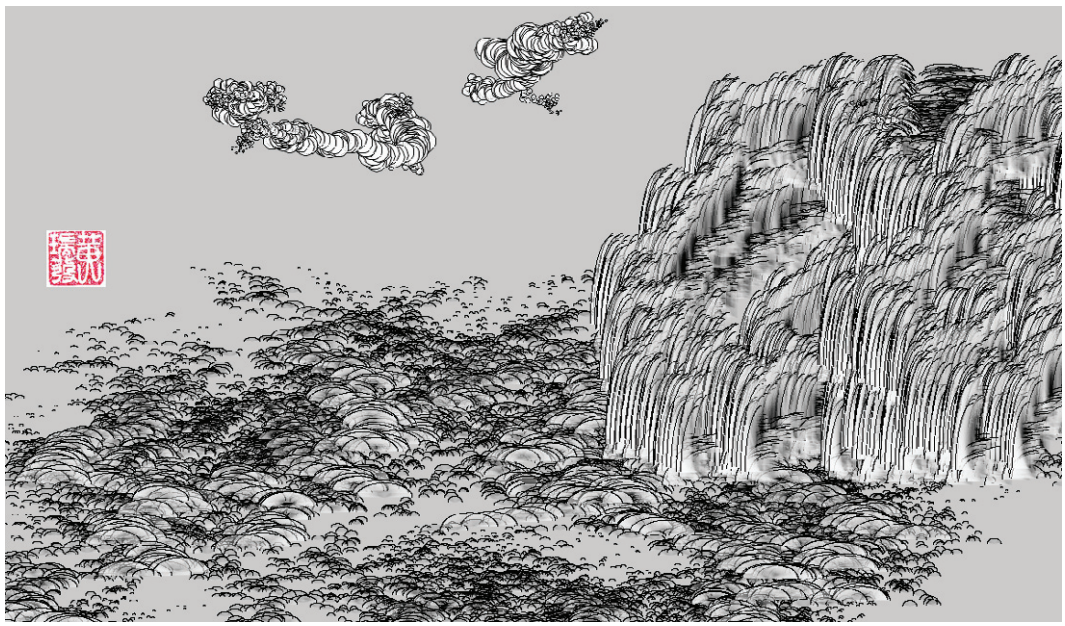
3.2. Incorporating Deterministic and Non-deterministic Control

Abstract figurative artifacts have been successfully generated using rules. Abstract 3D models and plant models have been successfully generated using mathematical models (see Figure 3 right-pane) and grammar (e.g.,

Trajectories of agents in an ecological system

```
process generate(w, h, e, o)
// Input parameters: width (w), height (h), energy (e), maxOffspring (o).
// Initialise n agents ag[n] and randomly place them in an imaginary 40x40 pixels
// box in the middle of the canvas. The size and colour of strokes are randomly assigned
// for each agent.
while at least one agent is still alive do
   for each living agent n
      if ag[n].produceOffspring()
         n ← n + 1;
         add new agent ag[n] to the pool
         break;
      endif
      ag[n].updatePosition(); // calculate the new position
      if ag[n].collidesWithOtherAgents()
         terminate all agents involved in the collision
         break;
      else
         ag[n].drawTrajectoryOnCanvas();// draw the path
      endif
      // Update parameters of agents and environment
      // e.g., reduce agent's energy by 1 unit; kill agents that have depleted the energy, etc.
      ag[n].updateParameters();
   endfor
endwhile
```

Fig. 4. Top: An abstract image generated from traces of particles. There is no explicit control, the behaviours of particles emerge from particles' interactions in the eco-system. Bottom: The pseudo code gives an overview of the main process employed to generate this image.



Incorporating deterministic and non-deterministic control

process generateCloud(mxd)
// Input parameters: max diameter (mdx)
// Initialise coordinate (x,y), diameter (D) and control-flag (stage)
$(x, y) \leftarrow initPoint(); D = 1; stage =$ "*rise*"
while not(stage == done) **do**
  // draw a circle centering at (x,y) with diameter = D pixels
  $circle(x, y, D)$;
  // stage control using rules and move the next (x,y) point randomly
  if ($stage ==$ "*rise*") $D = D + 0.1; x = x + random(-10, 10)$; **endIf**
  if ($stage ==$ "*decay*") $D = D - 0.1; y = y + random(-5, 5)$; **endIf**
  if ($stage ==$ "*rise*"&&$D >= mdx$) $stage =$ "*decay*"; **endIf**
  if ($stage ==$ "*decay*"&&$D <= 0$) $stage =$ "*done*"; **endIf**
endwhile

process generateWave(m, w, h)
// Input parameters: motive (m), width (w) and height (h) of the area to be drawn;
// Initialise n agents $ag[n]$ and place them on the top of the drawing area. Each
// agent will be randomly placed at (x,y) and randomly initialised with a scale (sc)
while at least one agent is still alive **do**
  for each living agent n
    if ag[n] is still in the canvas
      $ag[n].drawWaveMotive(x, y, sc)$;
      // Control the direction of the wave by adjusting the random points.
      $x = x + random(-5, 10)$;
      $y = y + random(-25, 25)$;
    endif
    // Update parameters of agents and environment: increase scale value, kill agents
    // that have moved outside the canvas or have reached the maximum scale, etc.
    $ag[n].updateParameters()$;
  endfor
endwhile

Fig. 5. Top: A landscape generated through repetition of simple motives. Bottom: The pseudo code showing examples of two generative processes employed to generate this image. To create an illusion of perspective, the motive will be scaled down to 10% of its original size at the start and linearly scaled up toward 130% at the end.

rewrite rules in the L-system [26]). Here, three landscape paintings are generated by exploiting deterministic and non-deterministic control embedded in the programs. Figures 5, 6, and 7 are landscape paintings generated using combinations of production rules and stochastic process. In our opinion, these are successful examples since figures could successfully represent world concepts (e.g., rock, wave, sun, star, sunflower, etc).

Figure 5, is drawn using motives derived from a circle. Most people see this image and perceive it as a landscape of offshore scenery with two lumps of cloud at the top and a cliff on the right hand side. The three components in this image are created using three simple functions that exploit both stochastic process and rules. Deterministic and non-deterministic control are exploited in this example, each cloud is drawn at a random position but constrained to the top half area in the figure. Once started, its size is randomly determined. The circular motives are repetitively drawn where each new circle is randomly positioned to the left or right of the previous circle. The diameter of the drawn circles starts at a minimum value, gradually increases to a maximum value (randomly determined) and reduces to the minimum again before the termination of the drawing process. In a similar manner, the wave and the cliff are drawn based on the same tactic. The pseudo code described in Figure 5 gives more detailed information of the control employed in generating the wave and the cloud components in this drawing.

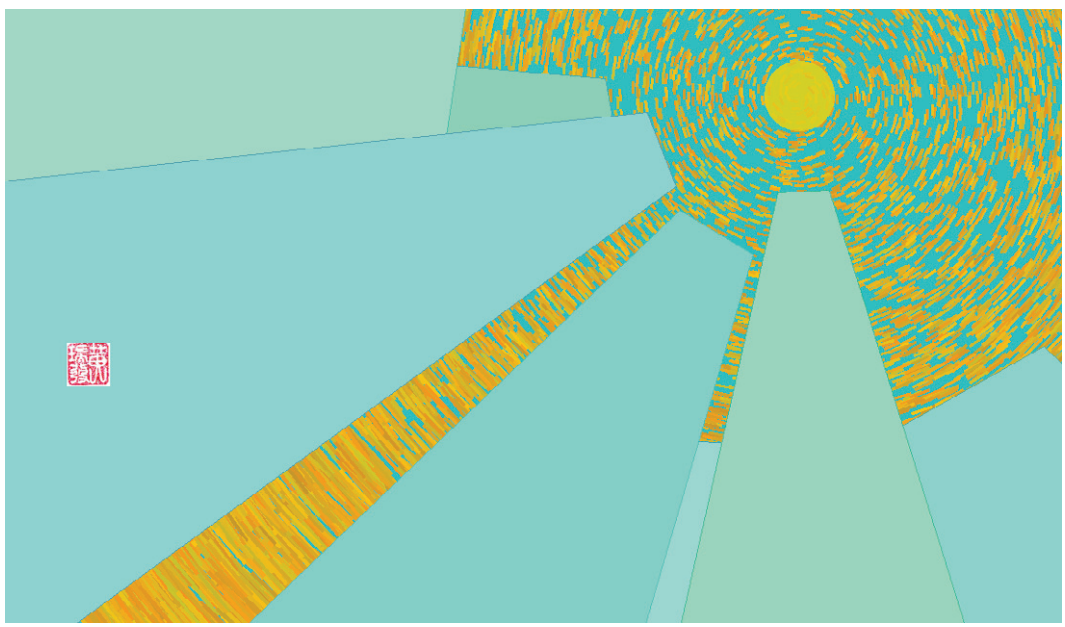


Fig. 6. An abstract figure generated using motive from a line. By associating this with our experiences of colours, figures and perspective, most people perceive this as an abstract landscape of an urban skyscraper.

Figure 6 is drawn using motives derived from a line. Most of us perceive this as looking up skyscrapers in an urban area. The sun rays are just short lines randomly drawn in a circle. The size of the line motive is increased the further away they are from the center point. This creates an illusion of depth. Figure 7 is the most complex drawing of all the examples shown in this paper. Inspired by the work titled *Sunflower field* by Fowler et al. [27], we created the model of a sunflower based on Vogel's formula [28]. The sunflower model can be modified (e.g., varying the number of petals, tilting the head at different angles). We then drew many rows of sunflowers and created an illusion of a sunflower field by controlling the position and the size of these sunflowers. The picture was composited with a dark blue random brush stroke as a background and twelve stars were randomly added to the final composition.

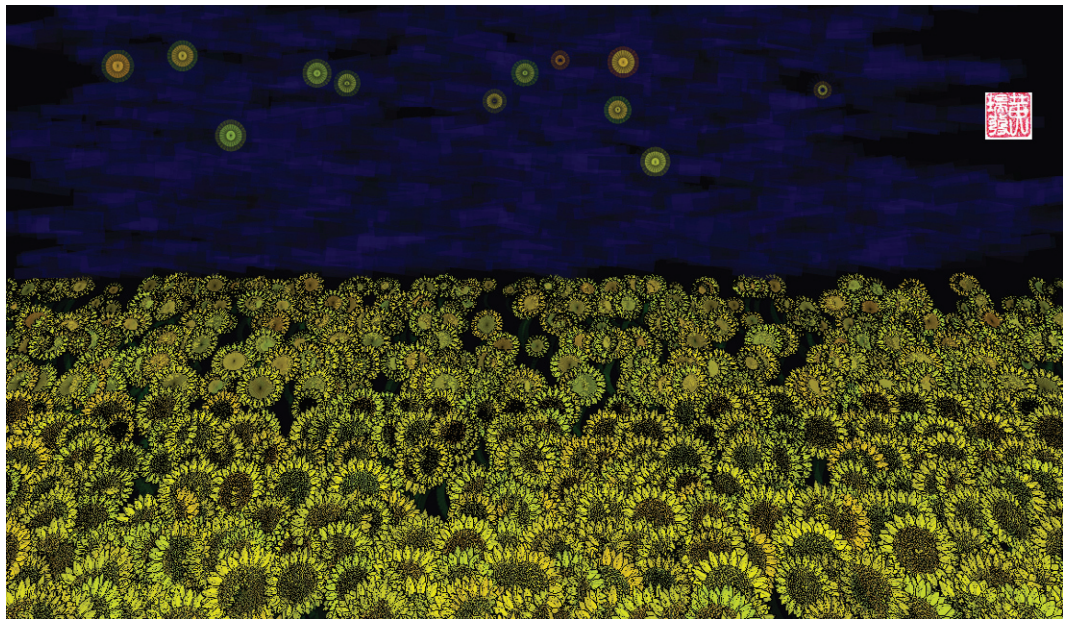


Fig. 7. The night scene of the sunflower field: the image is generated by drawing various variations of sunflower heads e.g., different number of petal, tilting at different angles, etc.

4. Conclusion

We investigated computational generative art systems where creative processes are implemented as computer programs. Ideally, the autonomous creative process should be able to explore the creative space independent of human intervention, as well as to justify which parts of the creative space are more fruitful. This can only happen if the system possesses a high level of creative intelligence which is unattainable at this point in time.

Contemporary generative art systems exploit computational techniques and computing tools (programing language) in generating aesthetic artifacts. Conditioned by the choices of knowledge representation and the mechanisms of knowledge construction, the contemporary generative art systems are mostly implemented using mathematical equations, stochastic process, evolutionary computing, swarm intelligence, grammar and rule-based techniques. Majority of the works focus on generating abstract visual artifacts and investigating aesthetic measures. Only the AARON and the L-system have attempted to generate non-abstract artifacts.

Generating non-abstract artifacts often requires a more complex program than generating highly ordered or highly disordered artifacts. This is because the relationships among components in non-abstract artifacts must hold together with coherent interpretations e.g., drawing a building: its windows and doors must be in suitable positions; drawing a sunflower: its stem, leaves and pedals must be within the right sizes and composited in a suitable composition, etc. Hence, the ability to exert control over the generative process is desirable and is very important in generating this kind of artifacts. Unfortunately, the issue of control has been somewhat neglected by the community. In this paper, four paintings were generated as case studies where the first example resorted to ecosystemic approach and the other three examples employed stochastic process and rules to control their generative processes. In our future work, we hope to further explore this control issue so that program behaviours can be compactly represented (as in the L-system) for generating various non-abstract artifacts.

Acknowledgements

We wish to thank anonymous reviewers for their comments and suggestions. We would also like to thank Mr. Adam Hardy for kindly proofreading the article.

References

[1] P. Galanter, What is generative art? complexity theory as a context for art theory, in: Proceedings of International Conference on Generative Art, Milan, Italy, 2003.

[2] M. Csikszentmihalyi, Creativity, in: R. Wilson, F. Keil (Eds.), The MIT Encyclopedia of the Cognitive Sciences, The MIT Press, pp. 205–206.

[3] F. H. George, Philosophical Foundations of Cybernetics, Abacus Press, 1979.

[4] M. Boden, Creativity and computers, in: T. Dartnall (Ed.), Artificial Intelligence and Creativity: An Interdisciplinary Approach, Kluwer Academic Publishers, 1994, Ch. Prologue, pp. 3–26.

[5] G. Wallas, The Art of Thought, New York: Harcourt-Brace, 1926.

[6] S. Dasgupta, Creativity, invention and the computational metaphor: Prolegomenon to a case study, in: T. Dartnall (Ed.), Artificial Intelligence and Creativity: An Interdisciplinary Approach, Kluwer Academic Publishers, 1994, Ch. Prologue, pp. 309–324.

[7] J. Gero, Computational models of creative design processes, in: T. Dartnall (Ed.), Artificial Intelligence and Creativity: An Interdisciplinary Approach, Kluwer Academic Publishers, 1994, Ch. Prologue, pp. 309–324.

[8] G. Wiggins, A preliminary framework for description, analysis and description of creative systems, Journal of Knowledge-Based Systems 19 (7) (2006) 449–458.

[9] S. Hélie, R. Sun, Incubation, insight, and creative problem solving: A unified theory and a connectionist model, Psychological Review 117 (8) (2010) 994–1024.

[10] S. Phon-Amnuaisuk, Control language for harmonisation process, in: Music and Artificial Intelligence, LNAI2445, Edinburgh, Scotland, UK, 2002, pp. 155–167.

[11] G. Flake, The Computational Beauty of Nature, The MIT Press, 1998.

[12] P. Machado, J. Romero, A. Santos, A. Cardoso, A. Pazos, On the development of evolutionary artificial artists, Computers & Graphics 31 (2007) (2007) 818–826.

[13] J. Rigau, M. Feixas, M. Sbert, Informational aesthetics measures, IEEE Computer Graphics and Applications (2008) 24–34.

[14] E. den Heijer, A. Eiben, Comparing aesthetic measures for evolutionary art, in: Applications of Evolutionary Computing, LNCS6025, Istanbul, Turkey, 2010, pp. 311–320.

[15] A. Ekárt, D. Sharma, S. Chalakov, Modelling human preference in evolutionary art, in: Applications of Evolutionary Computing, LNCS6625, Torino, Italy, 2011, pp. 303–312.

[16] T. Unemi, Sbart 2.4: an iec tool for creating 2d images, movies, and collage, Leonardo 35 (2) (2002) 189–191.

[17] A. Dorin, A survey of virtual ecosystems in generative electronic art, in: J. Romero, P. Machado (Eds.), The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music, Springer, 2008, Ch. 14, pp. 289–309.

[18] J. McCormack, O. Brown, Life's what you make: Niche construction and evolutionary art, in: M. Giacobini et al. (Ed.), Applications of Evolutionary Computing, Springer, 2009, pp. 528–537.

[19] H. Read, Modern Painting, Thames & Hudson, 1974.

[20] A. Lindenmayer, Mathematical models for cellular interaction in development, Parts I and II, Journal of Theoretical Biology 18 (1968) 280–315.

[21] P. Bentley, An introduction to evolutionary design by computers, in: P. Bentley (Ed.), Evolutionary Design by Computers, Morgan Kaufmann Publishes, Inc., 1999, Ch. 1-73, pp. 221–250.

[22] S. Todd, W. Latham, The mutation and growth of art by computers, in: P. Bentley (Ed.), Evolutionary Design by Computers, Morgan Kaufmann Publishes, Inc., 1999, Ch. 9, pp. 221–250.

[23] H. de Garis, Artificial embryology and cellular differentiation, in: P. Bentley (Ed.), Evolutionary Design by Computers, Morgan Kaufmann Publishes, Inc., 1999, Ch. 12, pp. 281–295.

[24] K. Sims, Evolving three-dimensional morphology and behaviour, in: P. Bentley (Ed.), Evolutionary Design by Computers, Morgan Kaufmann Publishes, Inc., 1999, Ch. 13, pp. 297–321.

[25] J. Bird, P. Husbands, M. Perris, B. Bigge, P. Brown, Implicit fitness functions for evolving a drawing robot, in: M. Giacobini et al. (Ed.), Applications of Evolutionary Computing, Springer, 2008, pp. 473–478.

[26] P. Prusinkiewicz, A. Lindenmayer, The Algorithmic Beauty of Plants, Springer, 1990.

[27] D. Fowler, N. Fuller, J. Hanan, A. Snider, Sunflower field, Interactions 5 (4), note Image reprinted in Interactions magazine, p.7 and p.49, 1998.

[28] H. Vogel, A better way to construct the sunflower head, Mathematical Biosciences 44 (1979) 179–189.